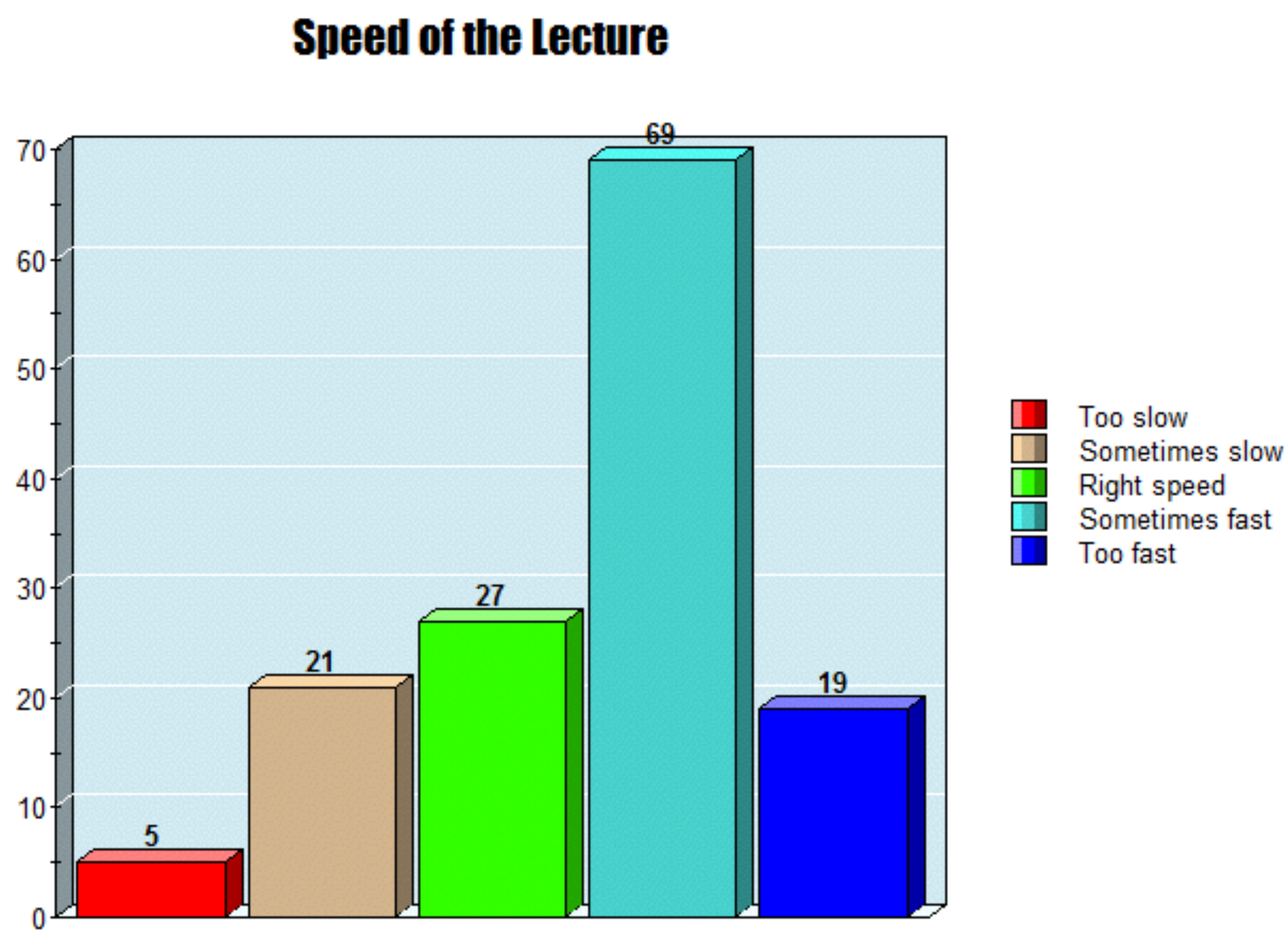
Machine Learning for Data Science (CS4786)
Lecture 11

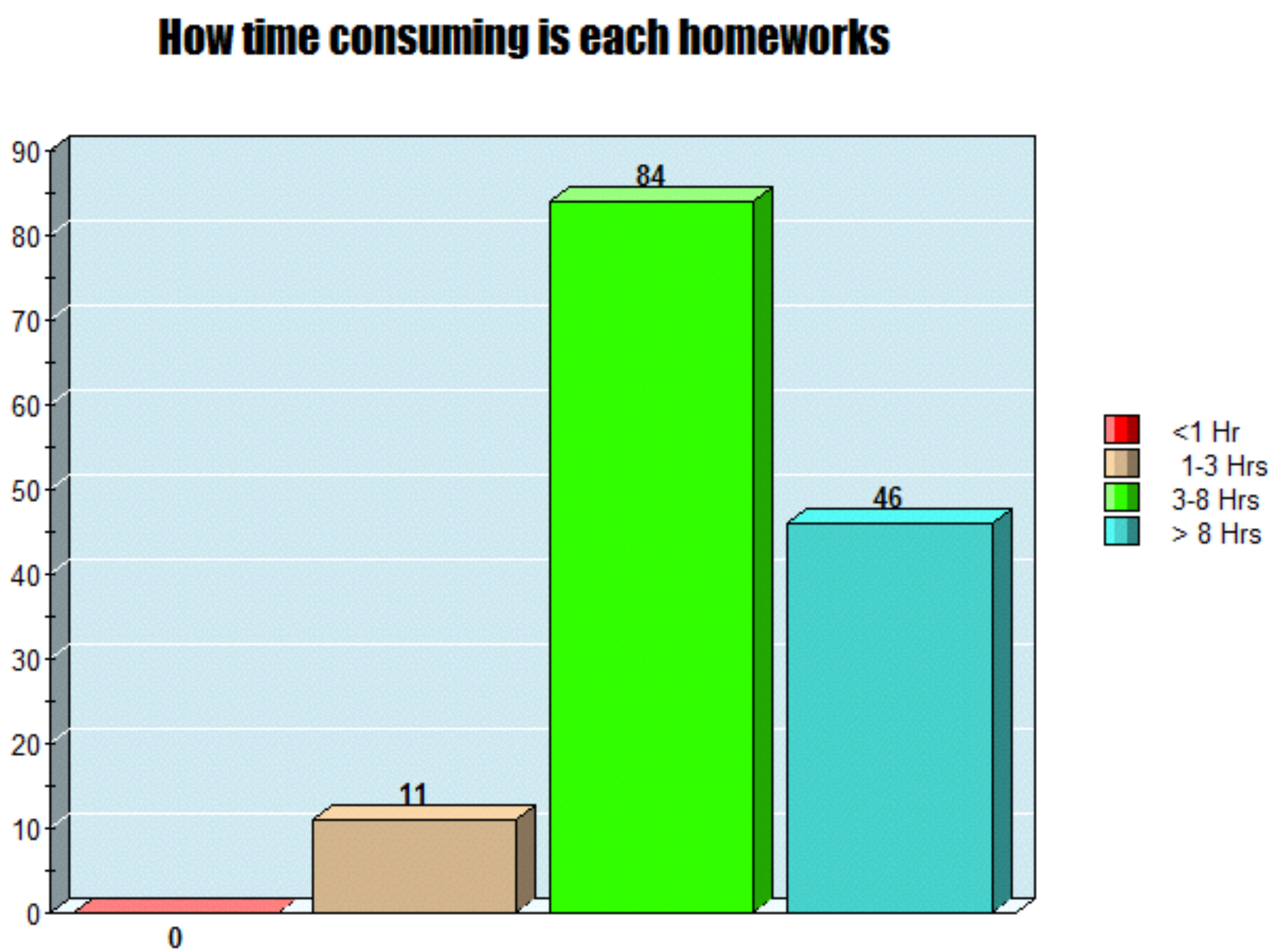
Spectral Clustering

Course Webpage :
http://www.cs.cornell.edu/Courses/cs4786/2016fa/

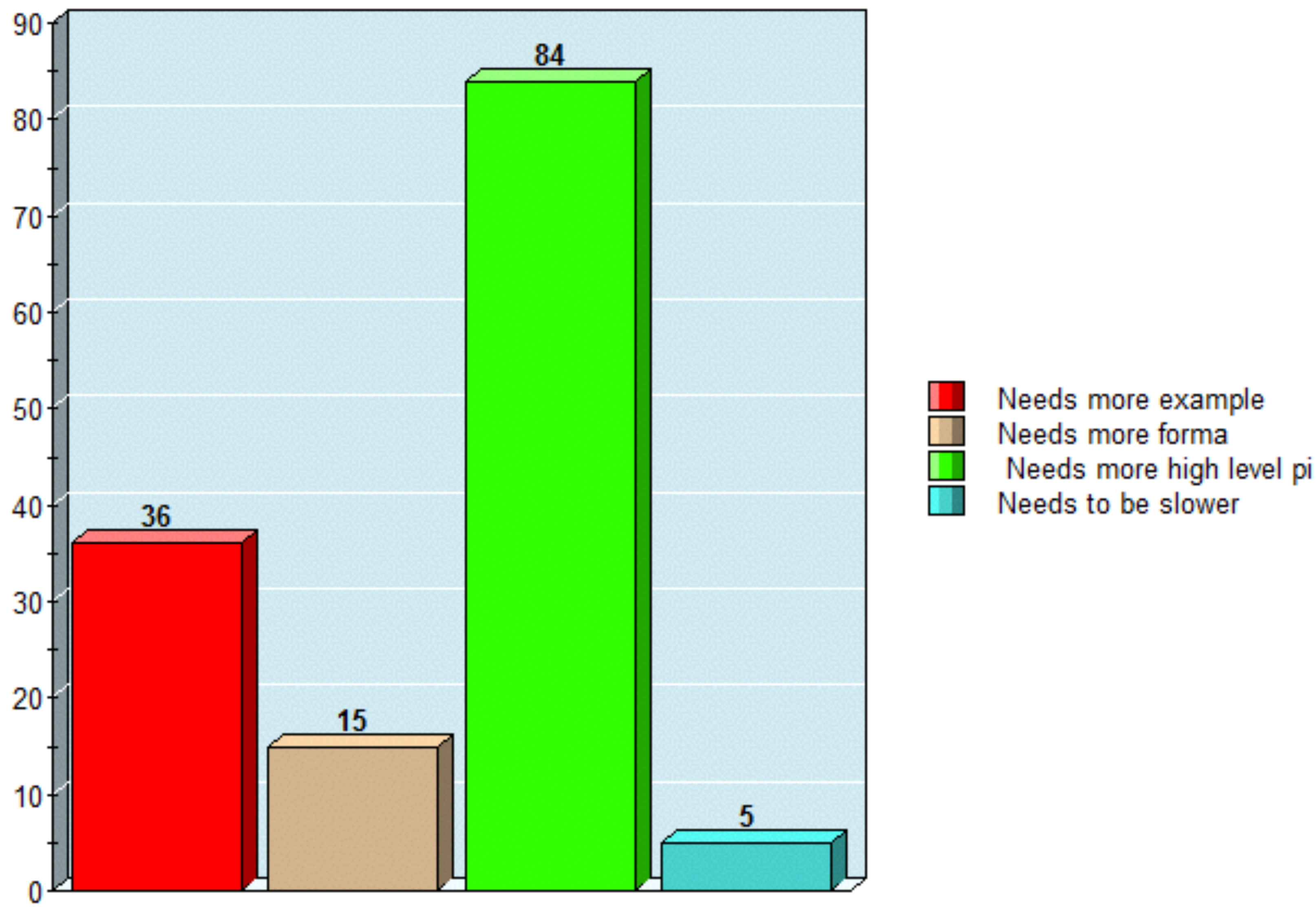
Survey


Speed of the Lecture
0
10
20
30
40
50
60
70
5
21
27
69
19
Too slow
Sometimes slow
Right speed
Sometimes fast
Too fast

Survey


How time consuming is each homeworks
90
80
70
60
50
40
30
20
10
0
84
46
11
0
<1 Hr
1-3 Hrs
3-8 Hrs
> 8 Hrs

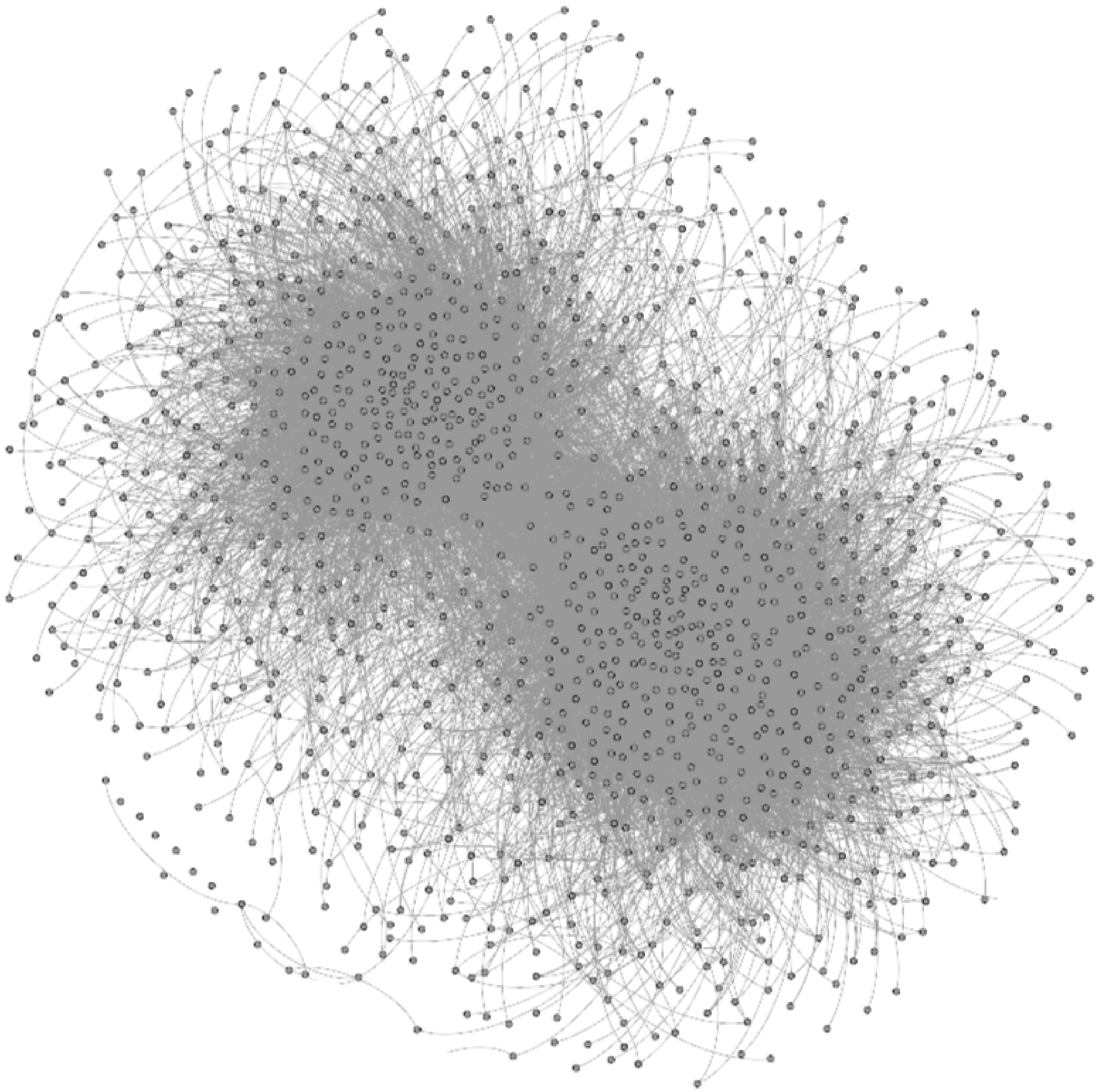
Survey

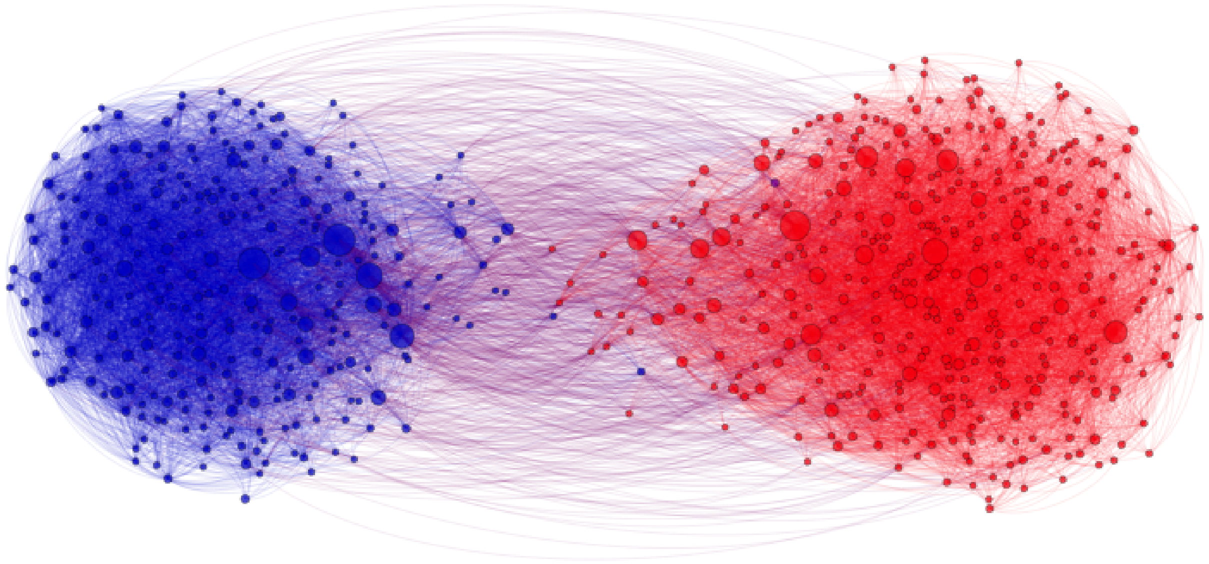

90
80
70
60
50
40
30
20
10
0
36
15
84
5
Needs more example
Needs more forma
Needs more high level pi
Needs to be slower

Competition I Out!

- Preliminary report of 1-2 pages due **Oct 4th**
  - Form your groups
  - Download data and familiarize yourself
  - Jot down preliminary ideas
  - In 1/2 page mention each group members contribution so far
- Competition closes **Oct 27th**

Spectral Clustering

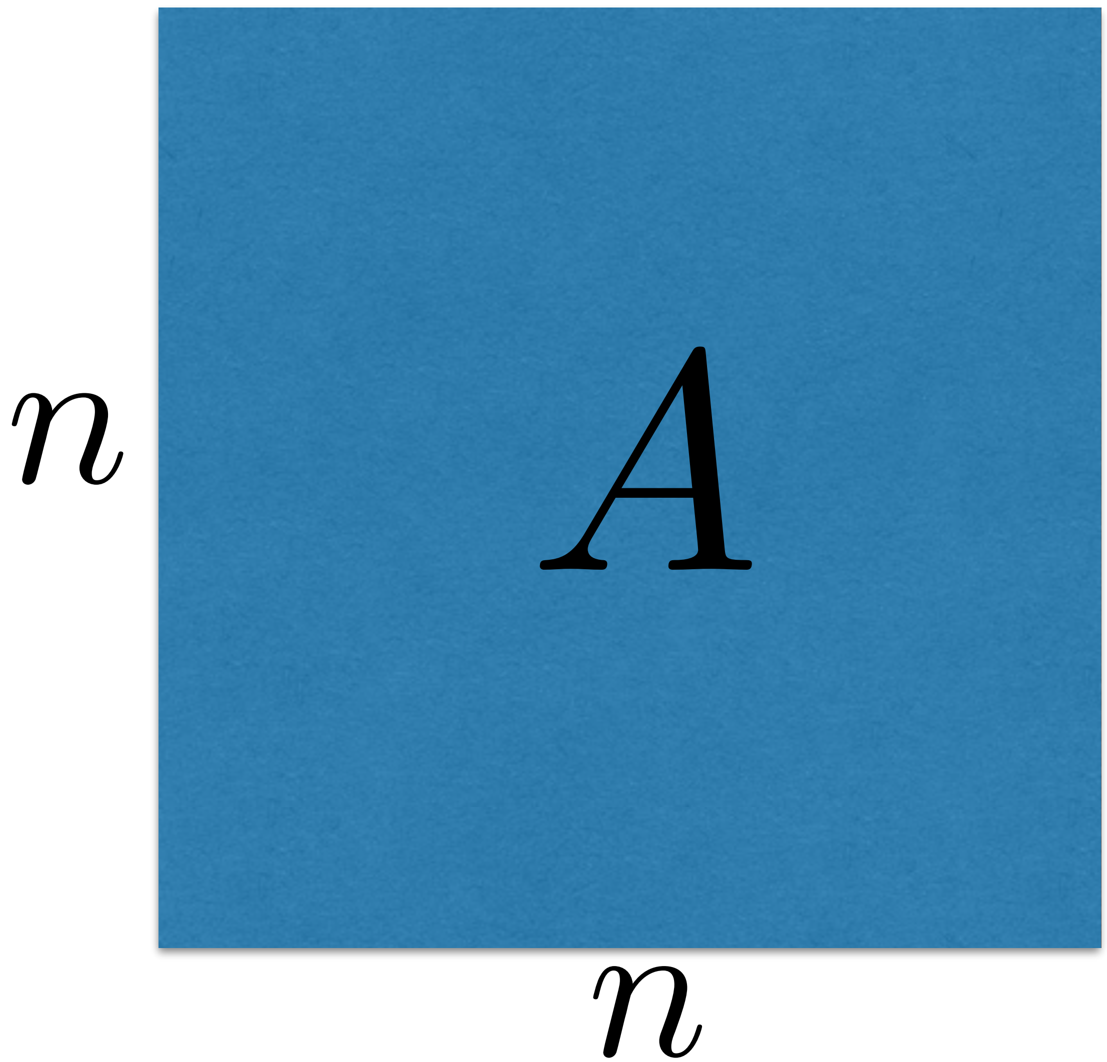




- Cluster nodes in a graph.
- Analysis of social network data.

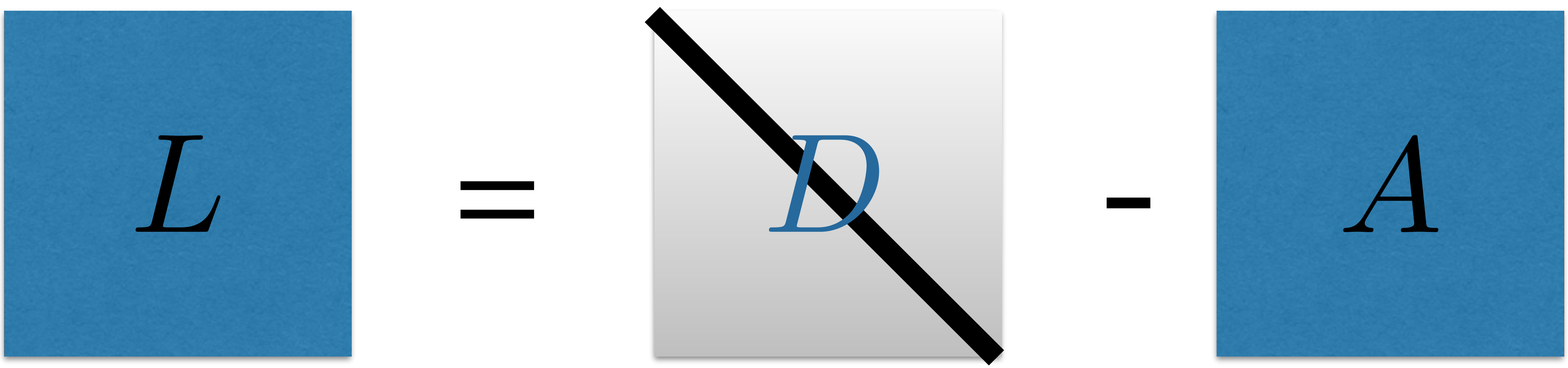
SPECTRAL CLUSTERING

$$A_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$



A is adjacency matrix of a graph

Spectral Clustering



$$D_{i,i} = \sum_{j=1}^{n} A_{i,j}$$

Spectral Clustering

$$\mathrm{Cut}(c) \sim \frac{1}{2} c^\top L c$$

Minimize $c^\top L c$ s.t. $c \perp \mathbf{1}$

Approximately minimize cut

Spectral Clustering Algorithm (Unnormalized)

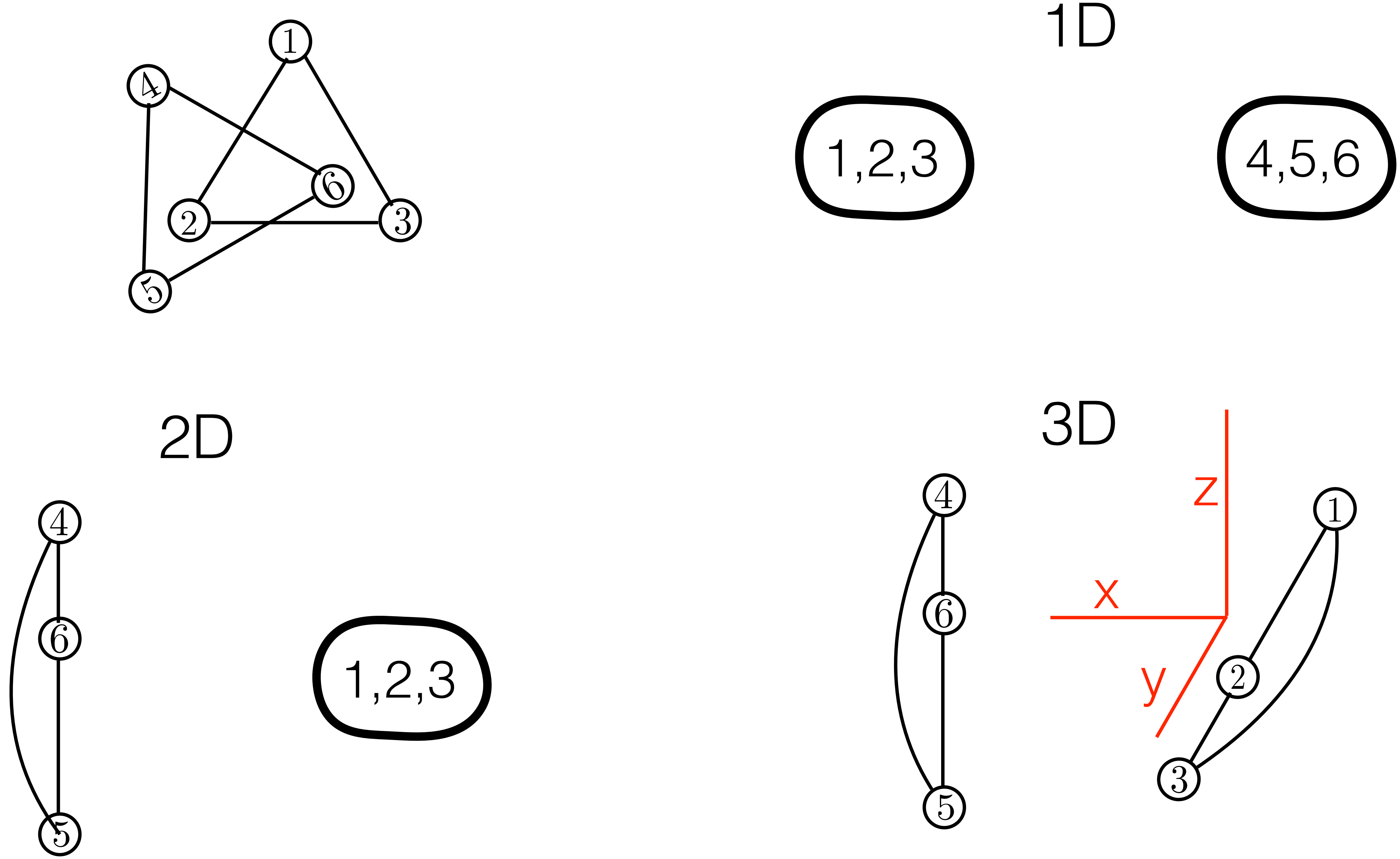
1. Given matrix A calculate diagonal matrix D s.t. $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$
2. Calculate the Laplacian matrix $L = D - A$
3. Find eigen vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of L (ascending order of eigenvalues)
4. Pick the K eigenvectors with smallest eigenvalues to get $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^K$
5. Use K-means clustering algorithm on $\mathbf{y}_1, \ldots, \mathbf{y}_n$

$\mathbf{y}_1, \ldots, \mathbf{y}_n$ are called spectral embedding

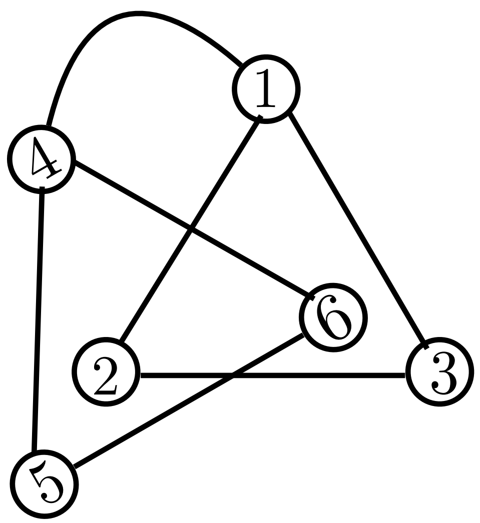
What is the Embedding?

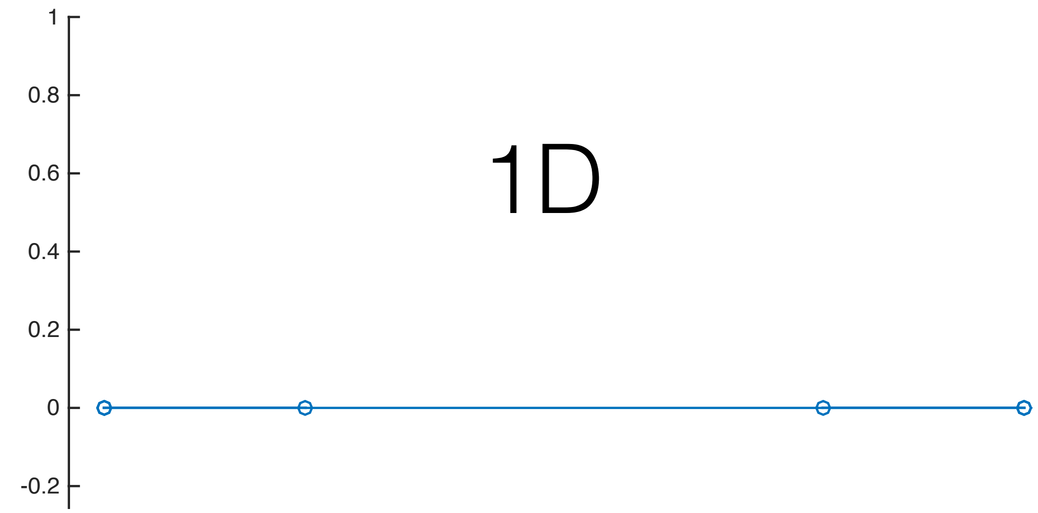
- Map each node in V to R^K
- Nodes lightly connected are farther
- Lets see some examples…

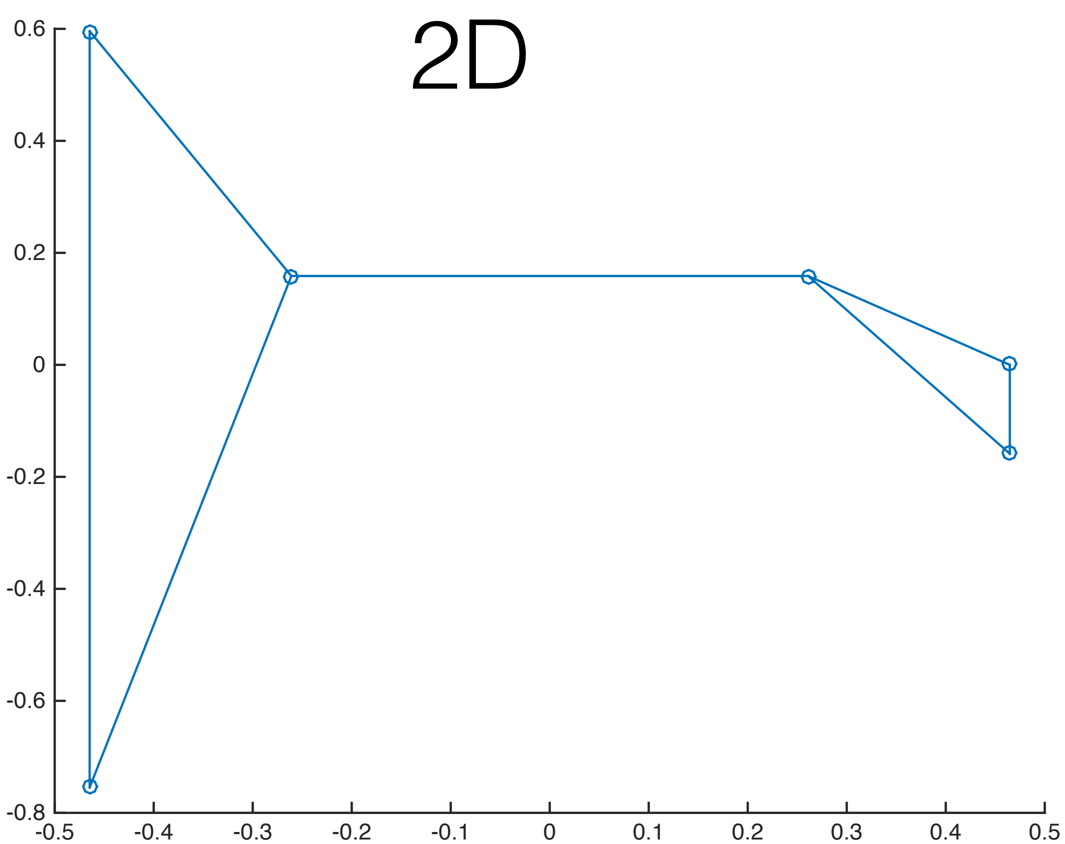
Examples

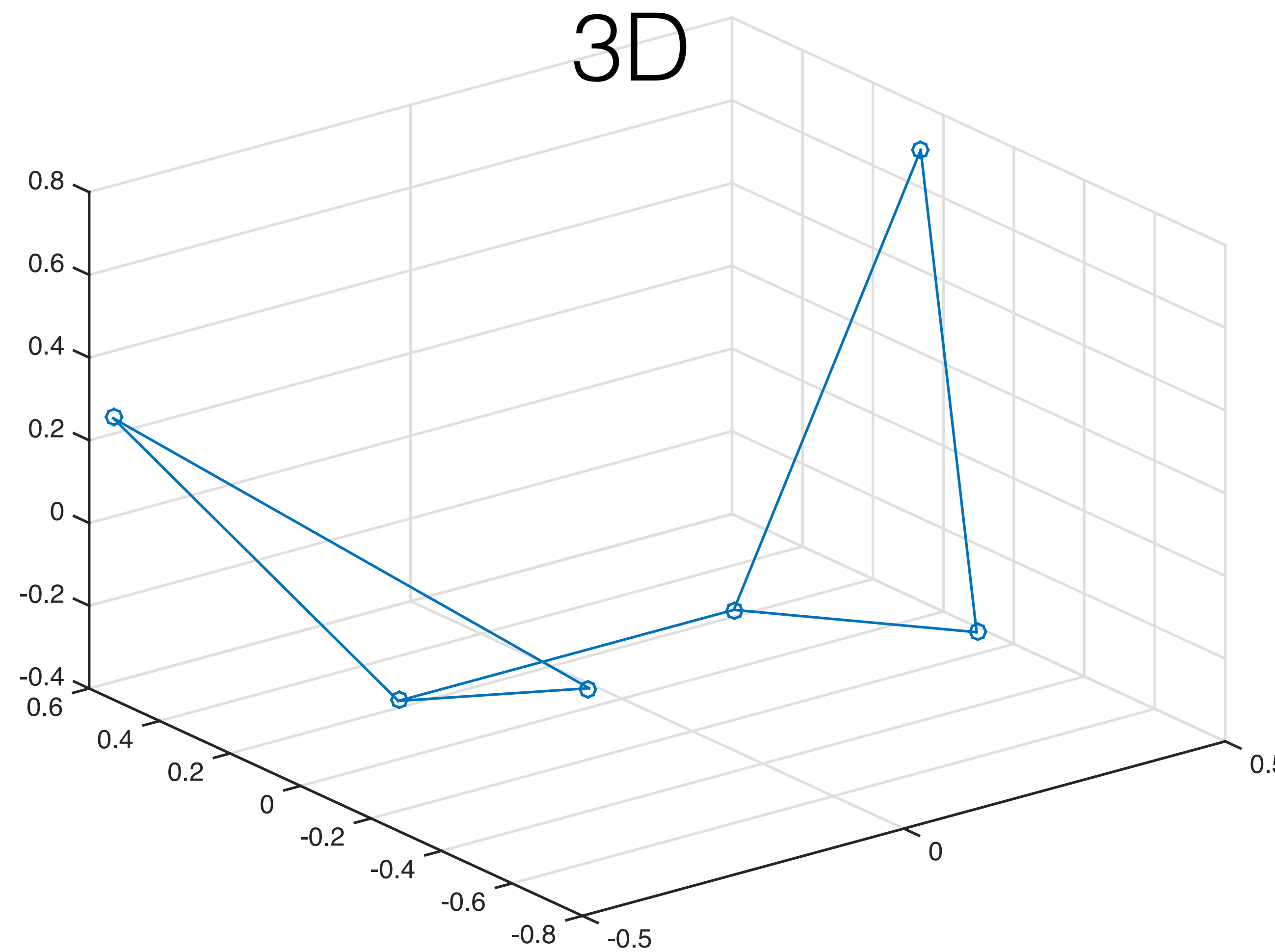


Examples


1
2
3
4
5
6


1D
1
0.8
0.6
0.4
0.2
0
-0.2


2D
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5


3D
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8
-0.5 0 0.5

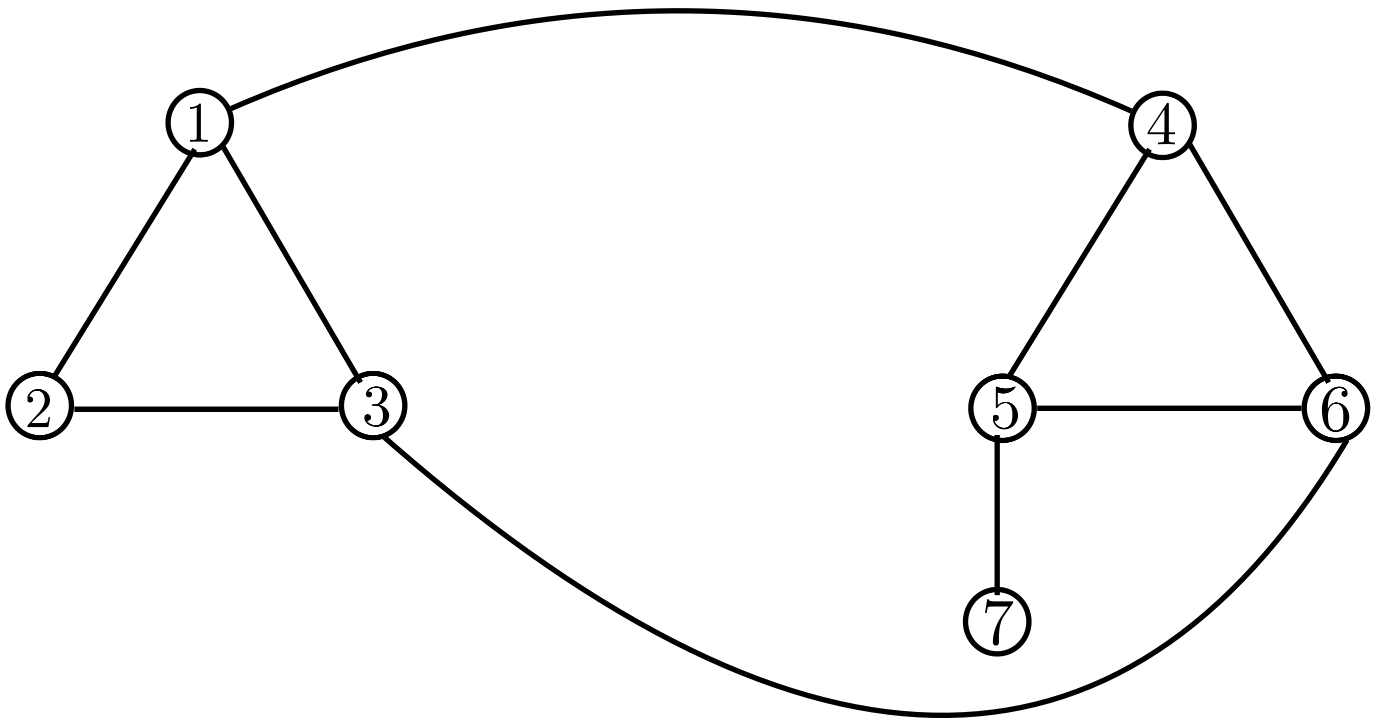
More Examples

Spectral Clustering (Unnormalized)

- Is cut even a good measure?

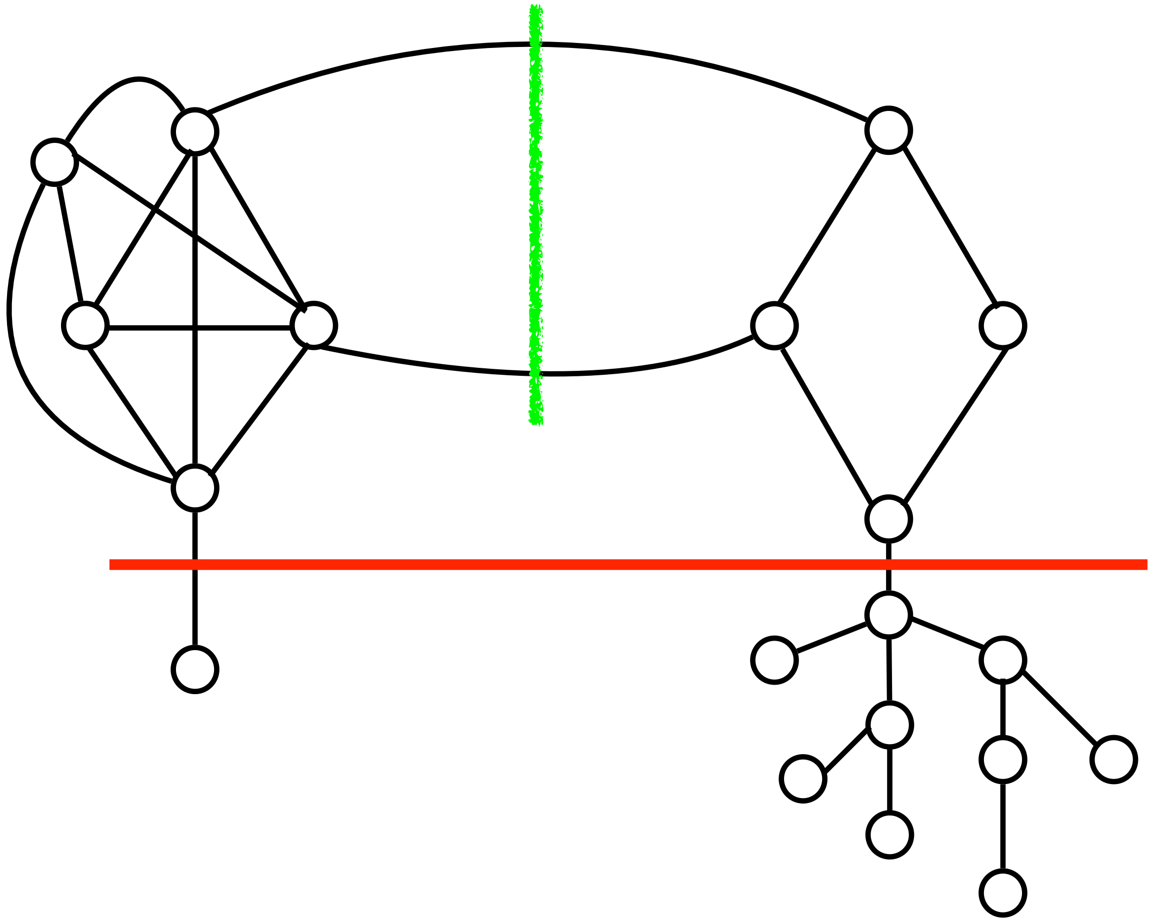
Ratio Cut

- Why cut is perhaps not a good measure?
- Fixes? Perhaps Ratio Cut : $\text{CUT}(C_1, C_2)\left(\frac{1}{|C_1|} + \frac{1}{|C_2|}\right)$



Ratio Cut

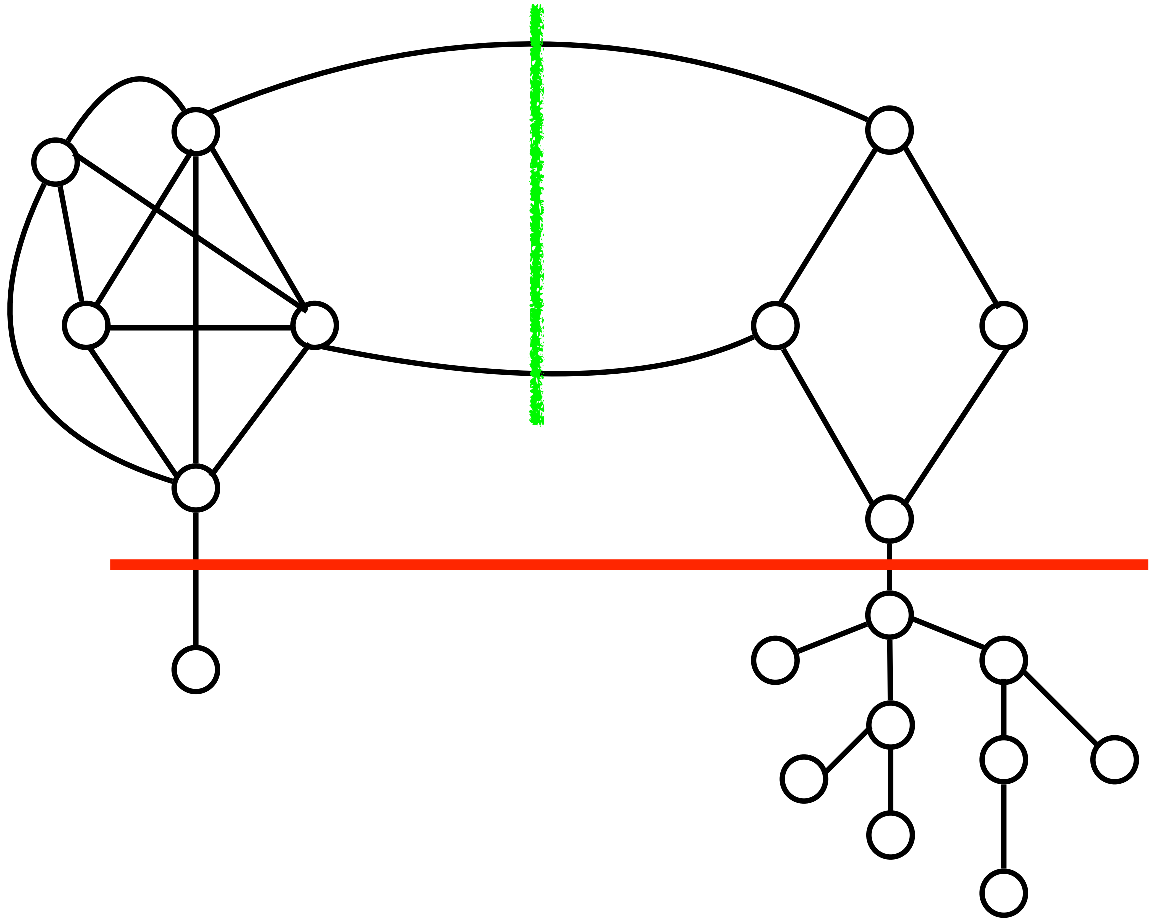
- Why cut is perhaps not a good measure?
- Fixes? Perhaps Ratio Cut : $\mathrm{CUT}(C_1, C_2)\left(\frac{1}{|C_1|} + \frac{1}{|C_2|}\right)$



Normalized Cut

- Normalized cut: Minimize sum of ratio of number of edges cut per cluster and number of edges within cluster

$$\mathrm{NCUT} = \sum_j \frac{\mathrm{CUT}(C_j)}{\mathrm{Edges}(C_j)}$$



$$\mathrm{Edges}(C_i) = \mathrm{degree}(C_i) = \sum_{t \in C_i} D_{t,t}$$

Normalized Cut

- Normalized cut: Minimize sum of ratio of number of edges cut per cluster and number of edges within cluster

$$\text{NCUT} = \sum_j \frac{\text{CUT}(C_j)}{\text{Edges}(C_j)}$$

- Example $K = 2$

$$\text{Edges}(C_i) = \text{degree}(C_i) = \sum_{t \in C_i} D_{t,t}$$

$$\text{CUT}(C_1, C_2)\left(\frac{1}{\text{Edges}(C_1)} + \frac{1}{\text{Edges}(C_2)}\right)$$

- This is an NP hard problem! ...so relax

Normalized Cut

- Set $c_i = \begin{cases} \sqrt{\frac{\text{Edges}(C_2)}{\text{Edges}(C_1)}} & \text{if } i \in C_1 \\ -\sqrt{\frac{\text{Edges}(C_1)}{\text{Edges}(C_2)}} & \text{otherwise} \end{cases}$
- Verify that $c^\top Lc = |E| \times \text{NCut}$ and $c^\top Dc = |E|$ (and $Dc \perp \mathbf{1}$)
- Hence we relax Minimize $\text{NCUT}(C)$ to

$$\text{Minimize } \frac{c^\top Lc}{c^\top Dc} \qquad \text{s.t. } Dc \perp \mathbf{1}$$

Spectral Clustering

$$\text{Minimize } c^\top \tilde{L} c \text{ s.t. } c \perp \mathbf{1}$$

Approximately Minimize normalized cut!

- Solution: Find second smallest eigenvectors of $\tilde{L} = I - D^{-1/2} A D^{-1/2}$

Spectral Clustering Algorithm (Normalized)

1. Given matrix A calculate diagonal matrix D s.t. $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$
2. Calculate the normalized Laplacian matrix $\tilde{L} = I - D^{-1/2} A D^{-1/2}$
3. Find eigen vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of $\tilde{L}$ (ascending order of eigenvalues)
4. Pick the K eigenvectors with smallest eigenvalues to get $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^K$
5. Use K-means clustering algorithm on $\mathbf{y}_1, \ldots, \mathbf{y}_n$

Demo

Normalized Cut: Alternate view

- If we perform random walk on graph, its the partition of graph into group of vertices such that the probability of transiting from one group to another is minimized
- Transition matrix: $D^{-1}A$
- Largest eigenvalues and eigenvectors of above matrix correspond to smallest eigenvalues and eigenvectors of $D^{-1}L = I - D^{-1}A$
- For K-nearest neighbor graph (K-regular), same as normalized Laplacian